HIGHLY SURVIVABLE AVIONICS SYSTEMS FOR LONG-TERM DEEP SPACE EXPLORATION

L. Alkalai, S. Chau, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove, Pasadena, CA 91109 leon@jpl.nasa.gov , Ann Tai, IA Tech Inc.

Introduction

The design of highly 'survivable' avionics systems for long-term (> 10 years) exploration of deep space is an essential technology for all current and future missions in the Outer Planets roadmap. Long-term exposure to extreme environmental conditions such as high radiation and low-temperatures make survivability in space a major challenge. Moreover, current and future missions are increasingly using available commercial technology such as deep sub-micron (0.25 μ) fabrication processes with specialized circuit designs, commercial logical interfaces, commercial processors, memory modules, and other COTS (commercial off the shelf) components that were not designed for long-term survivability in space. Therefore, the design of highly reliable, and highly available systems for the exploration of Europa, Pluto and the Kuiper Objects, and other destinations in deep-space require a comprehensive and fresh approach to this well known problem. This paper summarizes work in progress in three different areas:

1. A framework for the design of highly reliable and highly available space avionics systems.
2. Distributed Reliable Computing architecture
3. Guarded Software Upgrading (GSU) techniques for software upgrading during long-term missions such as the Pluto/Kuiper missions.

A Framework

Significant advances in the field of computing and information technology warrant a fresh look at the problem of designing highly reliable avionics systems. Missions such as Cassini use highly reliable and expensive components organized into sub-systems which are cross-strapped between a prime and a backup. Today's missions such as the Europa Orbiter use advanced computing technologies such as the PowerPC 750 and many other components that have a strong commercial heritage. As this trend continues, future deep-space missions will have to be designed to tolerate and survive higher number of failures, many of which will not be well understood as failure modes during the regular qualification process. Our work in progress is considering a comprehensive framework for the design of reliable systems that accommodates the use of advanced (and thus less known) technologies. This approach is based on a hierarchical, stratified methodology that starts from the physics of failure analysis at the materials and devices layer, and ranges to the hardware/software application or service layer. Reliable services are provided in a distributed fashion over a reliable network with a rich set of interconnections. Moreover, fault-detection, containment and recovery are handled in a localized fashion, with hierarchical coordination with a higher layer only if necessary. In the limit, at starting at a very fine granularity, such an approach is even consistent with biological systems, which can also serve as an excellent reference model.

Distributed Reliable Computing Architecture

Our work in progress has shown that a distributed system that consists of a redundant and scaleable network with redundant storage devices and computing nodes, can provide an effective platform for the implementation of reliable network services. This work, funded jointly by NASA and DARPA, resulted in a commercial spin-off called *RAINfinity* which has applied these techniques to reliable services over the internet. Our current design of a distributed system uses the 1394 Firewire network to provide for a rich set of redundant interconnects. Multiple nodes on the network can provide backup services for both high-reliability as well as high-availability (the latter is of great commercial interest).

Guarded Software Upgrading

Long-term missions such as Pluto/Kuiper Express will benefit greatly from the proposed technology we refer to as "Guarded Software Upgrading." It is expected that during a mission that lasts 10-14 years, there will be plenty of opportunities to routinely upgrade the software. Currently, there is no known simple and reliable way to continuously upgrade software. In fact, it is quite a difficult and risky prospect. Together with IA Tech Inc (and via several SBIR contracts), NASA has been developing techniques to reliably upgrade software. GSU allows the old and reliable version of the software to co-exist with the newly uploaded, and thus less reliable software. Using inherent redundancy (spare computer node), and distributed checkpoint and rollback techniques, the old software 'guards' the transition to the new software in a reliable fashion, until the right level of confidence is reached. At that point, the new software assumes control. A working prototype of this exciting new technology has been demonstrated, and is now in the latter stage of maturity.

Acknowledgement:

The research described in this paper was performed at the Center for Integrated Space Microsystems, JPL,

California Institute of Technology, and sponsored by
the National Aeronautics and Space Administration.